Automatic Load Shedding calculated with Genetic Algorithms – DAC-CMAG

Michelle Guichon, Magdalena Melo, Ana Carolina Nieto, Mario Vignolo, Nicolás Yedrzejewski

Abstract-- **This paper presents an optimization tool based on Genetic Algorithms, DAC-CMAG (Automatic Load Shedding Calculated Through Genetic Algorithms), developed in Matlab and applied to the calculation of load shedding in Electric Power Systems. This application calculates the optimal load shed necessary to eliminate overloading of any series element of an electrical network. It includes a module that runs DC load flow to calculate the power flow for each branch or transformer and verifies there are no current violations in any equipment. The results are analyzed using this tool applied to the calculation of optimum load shed required for the worst contingencies in the 500kV power system of Uruguay.**

***Index Terms*—DC load flow, load shedding, Genetic Algorithm**

I. Introduction

IN electric power systems, electric energy demand is in general concentrated in cities and towns. On the other hand, generation may be far from any city, often responding to the type of primary energy source used by generating plants. The clearest example is made up of hydroelectric power plants that use the reservoir created by a river dam, transforming the gravitational potential energy of water to drive a turbine and generate electricity. This resource …

To transport energy from points of generation to demand, high and extra high voltage transmission networks are used. In Uruguay, high and extra high voltage comprises voltages up to 150 kV and 500kV respectively. Because of the importance of the transmission network, the power system is in general designed and built to be reliable and allow system operation even in the absence of one or more constituent elements, namely: lines, underground cables, transformers, etc. However, there are many contingencies for which it is necessary to disconnect part of the load to ensure the stability of the entire system and the integrity of the equipment. In most cases, disconnection should be performed so rapidly that it is possible only automatically, having chosen the stations and conveniently disconnections in advance.

The project includes the study of the behavior of genetic algorithms (GA) to optimally calculate the least amount of load that is necessary to disconnect from the electrical power system after a contingency has occurred, to ensure that any equipment be loaded beyond its load limit in an emergency.

II. DC Load Flow

DC load flow is implemented considering the common simplifications to load flow, namely:

- differences between bus angles sufficiently small:
  $\cos(\delta_i - \delta_j) = 1$
  $\sin(\delta_i - \delta_j) = (\delta_i - \delta_j)$
- shunt susceptance and series resistance not considered
- every bus in the system with constant 1pu voltage

The final equation of power flow for any series element is:

$$P_i = \sum_{n=1}^{N} B_{in} * (\delta_i - \delta_n)$$

Where B_{in} is the inverse of the series reactance of the branch between bus *i* and bus *n*.

III. Modeling of the Electrical Network

The electrical system is modeled in a simplified manner, using only the parameters necessary for the implementation of DC flows. The electrical system components included are: generators, lines and / or cables, transformers.

- The generators are modeled as a typical PV bus in any load flow, with the active power and voltage defined as known data. In this case the voltage, as in the rest of buses in DC load flows is 1 pu.
- In transformers, series resistance and the magnetizing branch are neglected leaving only the series reactance
- The load buses are modeled as constant active power regardless of the reactive power
- In the line model, series resistance, shunt conductance and shunt susceptance are neglected leaving only the series reactance.

IV. Genetic Algorithms

A. Introduction

Genetic algorithms are adaptive methods generally used in search and optimization data problems, based on sexual reproduction and the principle of survival of the fittest.

B. Adjustment function (fitness)

The fitness in the nature of an organism is defined as the probability that the organism survives to reproductive age and reproduces. The fitness function can be think as a numerical measure of profit, utility or goodness of a solution that can be maximize.

C. Operators

The simplest form of genetic algorithms involves three types of operations:

- Selection: select people in the population for reproduction. The selection of an individual is related to its adjustment value. The biggest adjustment has an inhabitant the more likely is to be chosen for reproduction. The easiest to create is a roulette wheel where each individual in the population is assigned a portion of the wheel, proportional to its fitness value. The better fitness has an inhabitant, the greater the portion of the wheel to be assigned, ergo, the more likely is to be chosen in the draw.
- Crossover: After being selected, individuals are crossed to produce offspring that are inserted into the next generation. This operator randomly chooses a crossover point and exchanges the subsequence before and after the crossover point between individual parents to form children.
- Mutation: This operator randomly changes one bit of the inhabitant given by the probability of mutation.

V. Fitness Function

It must be define a fitness function that qualifies each inhabitant of the population. After that, the population can be sorted by ranking.

The first thing to take into consideration is the load connected to the network. It is a desire to have connected load as much as possible. Therefore, the function must reward inhabitants with more load connected. For this, two variables are created to quantify the load remains connected, the first is the amount of load remains connected (*Pconectada in the software*), the second is the number of buses that remains on service (*cantbarras in the software*). The fitness function is implemented to be an increasing function in these variables:

$$fitness \propto Pconectada \times cantbarras$$

Another variable to consider is the overloaded equipments. The main target is to eliminate every current violation. Therefore, the function fitness must assign smaller fitness values for the most overloaded network.

In this case, two variables are defined to represent how overloaded is the network: *sob*, number of overloaded branches (or transformers) and *Smax*, is the ratio between the line load of the most heavily loaded line and the thermal capacity of the line. The following equation for the fitness function is proposed.

$$fitness = \begin{cases} Pconectada^3 \times cantbarras^3 \times (S_{\max} \times 100)^9 ; sob = 0 \\ \dfrac{1}{\left(e^{\frac{sob}{10}}\right)^2} \times \left(\dfrac{Pconectada^3 \times cantbarras^3}{S_{\max}{}^2}\right); sob \geq 1 \end{cases}$$

Where *sob,* is the number of overloaded branches in the system.

This function accomplishes all the requirements. For further details in the development of the function, see [1].

VI. Main program

This section discusses the implementation of major program that calculates the optimal load shedding for the various contingencies that may occur in the network using tools DC power flow and genetic algorithms.

Input and data procesing

The network data is read from an Excel spreadsheet as explained below. In the first page, called *Barras*, is the list of stations with their respective power loads and generation, which expressed in MW. In some cases, depending on user requirements, there would be convenient to leave certain fixed stations. This option is included on the first page where the user must put a 1 for allow the station of interest to participate in the load shedding and a zero in those buses not allowed to participate. In the second and third sheet there are the line and transformer data. They contain the impedance of each line, expressed in pu, and up to 2 maximum current limits (A and B), expressed in Amper. Also expresses the voltage base.

The network data is read when the file is opened through the graphical user interface of the program. The function devuelveDatos.m is who is responsible for processing them for later use in the rest of the program.

Then, functions DevuelveXY.m and armaMatrizA.m are executed. The first, return arrays *X* and *Y* corresponding to the series impedance and admittance of lines and transformers. The second returns a matrix whith a dimension *NxB* for a network with N lines B buses. For line *i*, $a_{ij}=1$ when j is the first bus of the line and $a_{ij}=-1$ for the second bus. Matrix *A* completely describes the topology of the network and is independent of the particular values of line parameters. Once the angles of the buses are calculated (vector *Ang*), the power flow throw the line from bus *i* to *j* is:

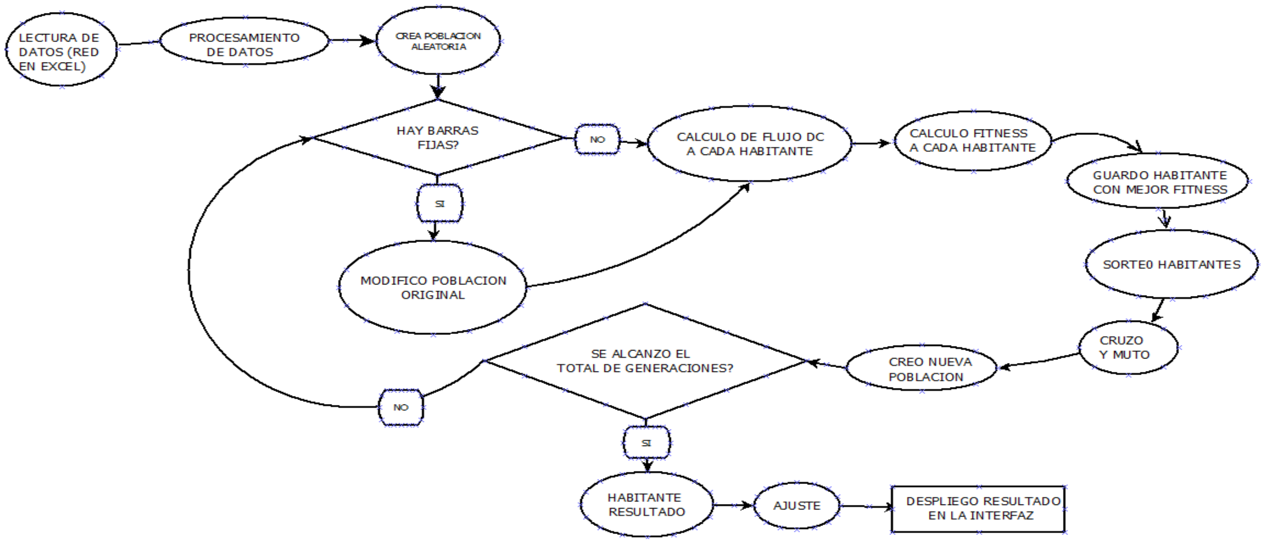
$$P_{ij} = A \times Ang \times Y$$

Main program

Initially, the program creates a certain amount of individuals to form the initial population. Each inhabitant represents a possible combination of substations to disconnect. Each individual is represented by a vector, with length equal to the number of substations in the network.

Each vector element is associated with a single substation and can take only two possible values, 0 or 1. The value 1 indicates that the substation remains in service and is not disconnected in that combination. The value 0 indicates that the substation is part of the load shedding for that combination.

Ultimately, each inhabitant represents the state of all network stations in the post-contingency situation. An iteration start, where in each step the program creates a new generation doing the following:



1. The program checks for fixed stations. If so, an amendment is created in the population by setting the value 1in positions corresponding to stations that must remain connected in every inhabitant.

2. Next, it calculates the fitness of each inhabitant of the population. To do this, function *flujoDChabitante.m* calculates a DC power flow to find the angle of every bus. In a common implementation of a DC power flow, a previews balance between load and generation is mandatory so there is no need of a slack bus like in AC power flows. However, in this case every inhabitant has different load connected so a correction in generation must be done before running a DC power flow. For this purpose, the first bus entered in the list of buses will be considered to state the balance.

3. Function elem.m calculates the node admittance matrix, Ybus. flujoDePotencia.m calculates the active power en every series element using matrices A and Ybus explain above.

4. Sobrecarga.m calculates the overloaded lines and transformes.

5. fitness.m calculates the fitness of every inhabitant and place in matrix habmax the one with best fitness for every generation.

6. To create the next generation, inhabitants are drawn, by sorteo.m function which uses ruleta.m (ruleta.m function is based on the selection operator of genetic algorithms explained above).

7. Function cruzaymuta.m cross and mutate the inhabitants drawn, creating a new population.

This process is repeated until the total number of generations is reached. The inhabitant of best fitness of all generations is taken as a partial solution.

For this inhabitant, function *ProgramaPrincipalAjuste.m* implements a final adjustment to the partial solution to ensure that the combination selected do not disconnect excessive and unnecessary load. The setting procedure is described below:

- The program goes through every bit of the inhabitant changing each zero by one and re-run the DC power flow. In each test, if the new inhabitant has no overloaded elements is stored in an array. The resulting matrix contains all the changes successful.
- The inhabitant of the matrix with lower disconnected load is chosen and the rest are discarded.
- With this new individual, the process is repeated.
- When every additional change is unsuccessful, the adjustment is terminated.

The adjusted inhabitant is considered the best solution found. Through the user interface, the program shows the final combination of station to be disconnected, the active power associated and the most overloaded line or transformer expected if the solution suggested is implemented.

VII. Applications in the Transmission Network of Uruguay

This chapter analyzes the behavior of the tool developed in this project to calculate the necessary load shedding for the most severe contingencies in the 500kV system of Uruguay.

The rate chosen is the thermal capacity the equipment supports for an hour and no more, which is appropriate in this regime. It considers a peak demand for the summer and winter, 2012.

It works with a mutation probability of 0.1%, 70% for crossover, population size of 1000 inhabitants and initially with 100 generations. This can vary according to the number of lines that may be overloaded.

PAL500-MVA500 and BRU500-MVB500 double contingency for summer peak

The following table shows the lines that are overloaded and the percentage of overload without load shedding:

Initial Bus	Final Bus	Current (A)	Overload (%)
ACO150	MVB150	951.603	183.001
BAY150	PAL150	-1101.14	229.418
BAY150	TRI150	559.068	155.348
COL150	ROS150	281.748	117.308
COL150	JLA150	567.059	136.666

CON150	COL150	928.72	258.063
DUR150	FLO150	1357.076	220.362
EFI150	SVA150	479.354	199.583
FBE150	MER150	975.182	203.175
FLO150	PRO150	1253.045	203.469
JLA150	LIB150	567.059	191.581
LIB150	EFI150	506.528	140.749
MER150	NPA150	1008.242	419.79
NPA150	CON150	931.804	224.573
PAL150	TRI150	1671.054	348.158
PAY150	YOU150	575.913	138.8
PRO150	MVA150	1253.045	203.469
ROD150	ACO150	1024.68	213.488
ROD150	MVB150	1210.878	195.888
SAL150	PAY150	741.251	154.437
SGU150	SAL150	890.733	185.581
SJA150	FBE150	1009.11	163.859
TER150	BAY150	-923.123	149.337
TER150	DUR150	1421.026	197.43
TRI150	ROD150	2192.466	228.396
SGU500	SGU150	306.131	160.677
SJA500	SJA150	302.733	158.894
PAL500	PAL150	832.608	277.331

To clear the overloads of the network the program calculates that 858 MW to be disconnected by shedding the following (all at 150kV):

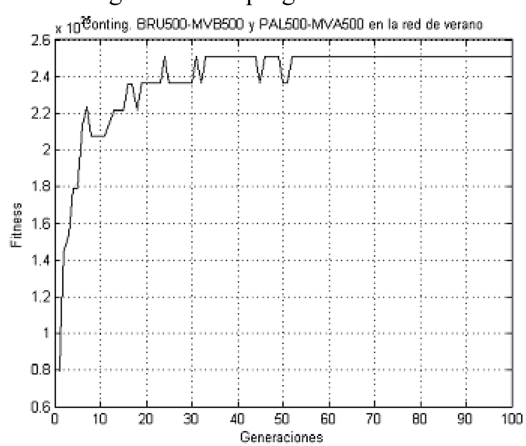
COL150	MEL150	MVE2	MVJ150	NPA150	PIE150
FLO150	MVA150	MVF150	MVK150	PAL150	ROC150
MAL150	MVC150	MVH150	NOR150	PES150	SOL150

After implementing the solution, the overloading lines are as follows:

Initial Bus	Final Bus	Current (A)	Overload (%)
ACO150	MVB150	195.268	37.551
BAY150	PAL150	-216.514	45.11
BAY150	TRI150	246.611	68.526
COL150	ROS150	102.917	42.85
COL150	JLA150	128.731	31.025
CON150	COL150	231.647	64.368
DUR150	FLO150	397.67	64.574
EFI150	SVA150	41.026	17.082
FBE150	MER150	305.566	63.663
FLO150	PRO150	397.67	64.574
JLA150	LIB150	128.731	43.492
LIB150	EFI150	68.2	18.951
MER150	NPA150	234.731	97.732
NPA150	CON150	234.731	56.572
PAL150	TRI150	478.268	99.645
PAY150	YOU150	20.069	4.837
PRO150	MVA150	397.67	64.574
ROD150	ACO150	268.345	55.909
ROD150	MVB150	283.141	45.805
SAL150	PAY150	185.406	38.629
SGU150	SAL150	334.889	69.773
SJA150	FBE150	339.493	55.127
TER150	BAY150	-350.953	56.775
TER150	DUR150	461.62	64.135
TRI150	ROD150	687.224	71.59
SGU500	SGU150	139.378	73.154
SJA500	SJA150	101.848	53.456
PAL500	PAL150	209.385	69.743

As can be seen, the most overloaded element, PAL150-TRI150, is a percentage of 99.645%.

The following shows that it took more than 50 generations to reach convergence of the program



PAL500-MVA500 y BRU500-MVB500 double contingency for winter peak

The following table shows the lines that are overloaded and the percentage of overload without load shedding:

Initial Bus	Final Bus	Current (A)	Overload (%)
ACO150	MVB150	1129.246	217.163
BAY150	PAL150	-1325.66	276.196
BAY150	TRI150	620.993	172.555
COL150	ROS150	311.871	129.85
COL150	JLA150	658.949	158.813

CON150	COL150	1049.124	291.519
DUR150	FLO150	1552.945	252.167
EFI150	SVA150	577.152	240.302
FBE150	MER150	1044.069	217.528
FLO150	PRO150	1435.25	233.055
JLA150	LIB150	658.949	222.627
LIB150	EFI150	605.369	168.213
MER150	NPA150	1131.633	471.165
NPA150	CON150	1052.041	253.551
PAL150	TRI150	1954.756	407.266
PAY150	YOU150	728.021	175.459
PRO150	MVA150	1435.25	233.055
ROD150	ACO150	1189.558	247.84
ROD150	MVB150	1421.159	229.905
SAL150	PAY150	914.647	190.563
SGU150	SAL150	1071.26	223.193
SJA150	FBE150	1250.631	203.077
SVA150	MVC150	462.474	111.46
TER150	BAY150	-1085.72	175.64
TER150	DUR150	1619.262	224.971
TRI150	ROD150	2534.399	264.016
YOU150	TER150	481.332	116.005
SGU500	SGU150	347.702	182.496
SJA500	SJA150	375.189	196.923
PAL500	PAL150	985.343	328.205

To clear the overloads of the network the program calculates that 1057 MW to be disconnected by shedding the following (all at 150kV):

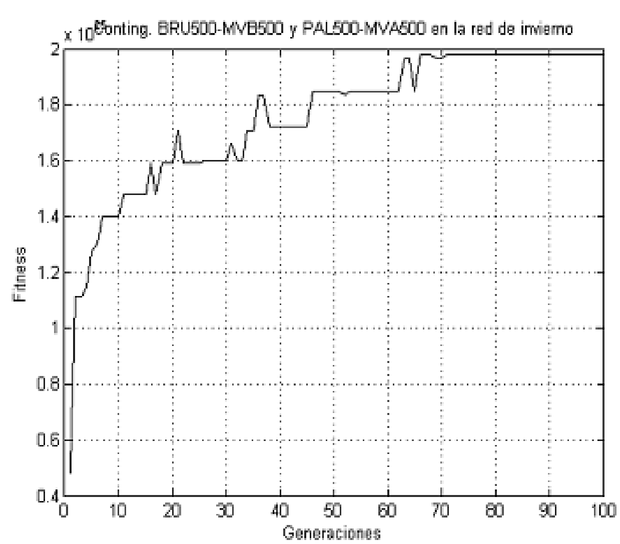
ACO150	MAL150	MVG150	PAN150	ROD150
COL150	MVC150	MVH150	PAZ150	ROS150
CON150	MVE2	MVR1	PIE150	SOL150
LIB150	MVF150	NOR150	RIV150	TRI150
				TYT150

After implementing the solution, the overloading lines are as follows:

Initial Bus	Final Bus	Current (A)	Overload (%)
ACO150	MVB150	324.216	62.349
BAY150	PAL150	-225.918	47.069
BAY150	TRI150	239.861	66.65
COL150	ROS150	2.129	0.887
COL150	JLA150	136.755	32.959
CON150	COL150	138.884	38.592
DUR150	FLO150	483.132	78.451
EFI150	SVA150	108.539	45.191
FBE150	MER150	234.055	48.764
FLO150	PRO150	365.437	59.34
JLA150	LIB150	136.755	46.203
LIB150	EFI150	136.755	38
MER150	NPA150	218.476	90.964
NPA150	CON150	138.884	33.472
PAL150	TRI150	479.917	99.989
PAY150	YOU150	46.01	11.089
PRO150	MVA150	365.437	59.34
ROD150	ACO150	324.216	67.549
ROD150	MVB150	397.691	64.336
SAL150	PAY150	232.636	48.469
SGU150	SAL150	389.249	81.099
SJA150	FBE150	440.618	71.547
SVA150	MVC150	-6.139	1.48
TER150	BAY150	-367.108	59.388
TER150	DUR150	549.449	76.337
TRI150	ROD150	719.778	74.981
YOU150	TER150	-97.537	23.507
SGU500	SGU150	143.098	75.107
SJA500	SJA150	132.185	69.379
PAL500	PAL150	212.968	70.937

As can be seen, the most overloaded element, PAL150-TRI150, is a percentage of 99,989%.

The following figure shows that it took more than 70 generations to reach convergence of the program.



PAL500-MVA500 y MVB500-MVA500 double contingency for winter peak

The following table shows the lines that are overloaded and the percentage of overload without load shedding:

Initial Bus	Final Bus	Current (A)	Overload (%)
MER150	NPA150	291.092	121.198
MVB150	MVL150	894.153	124.162
MVB150	MVA150	1267.857	132.077
MVD150	MVE2	776.173	107.78
COL150	JLA150	136.755	32.959
MVB500	MVB150	975.533	140.082

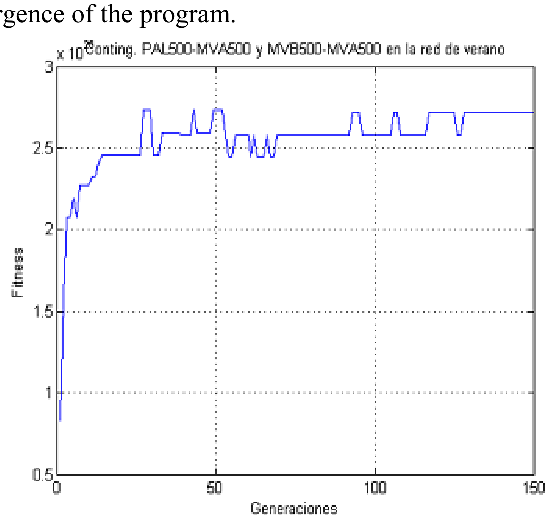
To clear the overloads of the network the program calculates that 347 MW to be disconnected by shedding the following (all at 150kV):

COL150	MVG150	PES150	TRI150
MAL150	MVL150	ROC150	
MVB150	MVR1	ROD150	

After implementing the solution, the overloading lines are as follows:

Initial Bus	Final Bus	Current (A)	Overload (%)
MER150	NPA150	186.315	77.574
MVB150	MVL150	636.31	88.358
MVB150	MVA150	895.24	93.26
MVD150	MVE2	528.031	73.323
COL150	JLA150	136.755	32.959
MVB500	MVB150	694.265	99.693

As can be seen, the most overloaded element, MVB500-MVB150, is a percentage of 99,693%. The following figure shows that it took more than 120 generations to reach convergence of the program.



PAL500-MVA500 y MVB500-MVA500 double contingency for winter peak

The following table shows the lines that are overloaded and the percentage of overload without load shedding:

Initial Bus	Final Bus	Current (A)	Overload (%)
MER150	NPA150	293.17	122.064
MVB150	MVL150	1077.838	149.669
MVB150	MVA150	1451.034	151.159
MVD150	MVE2	913.822	126.894
MVI150	MVM1	-999.556	107.745
MVM1	MVA150	-999.556	107.745
PAL150	TRI150	495.758	103.289
SGU150	SAL150	549.868	114.563
SGU500	SGU150	191.284	100.398
MVB500	MVB150	1140.638	163.791

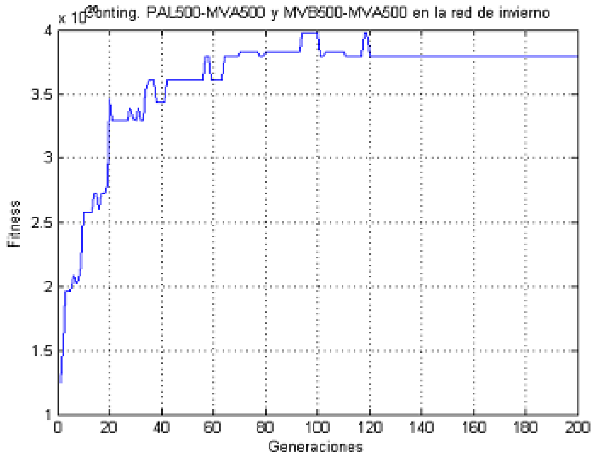
To clear the overloads of the network the program calculates that 529 MW to be disconnected by shedding the following (all at 150kV):

ACO150	MVC150	PAN150
EFI150	MVE2	PIE150
MAL150	MVF150	STE150

After implementing the solution, the overloading lines are as follows:

Initial Bus	Final Bus	Current (A)	Overload (%)
MER150	NPA150	161.021	67.042
MVB150	MVL150	706.96	98.169
MVB150	MVA150	950.007	98.965
MVD150	MVE2	597.457	82.963
MVI150	MVM1	-776.358	80.876
MVM1	MVA150	312.966	65.205
PAL150	TRI150	414.204	86.298
SGU150	SAL150	150.585	79.036
SGU500	SGU150	696.269	99.981

As can be seen, the most overloaded element, SGU500-SGU150, is a percentage of 99,981%. The following figure shows that it took more than 120 generations to reach convergence of the program.



VIII. CONCLUSIONS

In all cases the solution leaves a network which has no overloaded elements, and most loaded lines are very close to 100% capacity. While these two features are not sufficient conditions to ensure that the solution found corresponds to the optimal load shedding, are strictly necessary for the solution to be valid and were the premises of the project.

The result is achieved in a relatively short time (depending on the computer varies between 5 and 8 minutes). An exhaustive search program trying all possibilities would take months to run.

Part of the program's success is attributed to the dedication to the implementation of the fitness function in conjunction with the final adjustment to be made to the result found by genetic algorithms.